

Metis Project 5

For my project I'm working on Deep Reinforcement Learning in OpenAI's gym environment. I'll be taking the game environment 'breakout' from their library and teaching a computer agent to play it.

MVP

For my baseline, my goal is to teach the agent to be better at the game than someone pressing buttons randomly. I'll be following a tutorial that uses a simple algorithm, DQN, to assign values to actions. The tutorial doesn't have the full explanation of how to create it, but rather shows the skeleton of how to make it, so my need to have is to successfully create this agent that can play breakout relatively well. Ultimately, this need to have is to have set up a deep neural network that: 1) utilizes convolutional neural networks to let the agent 'see' the gamespace, and 2) utilizes fully connected layers to make the agent decide which action to take.

Having discussed it with Brendan, my original goal was to either 1) rewrite this code from scratch myself, or to 2) significantly refactor it myself. However, after having spent time tinkering with this tutorial, it's clear that the entire model isn't written out; rather, it gives a basic guideline to follow without actually creating the agent/making it work. As such, I think an MVP that is more feasible is just using this skeleton outline to write my own code to get the agent working/playing. This will also have the requirement that I'm able to speak to the code and explain what is happening along the way.

As a deliverable, I'd like to be able to have videos of my agent playing: a clip of it playing naively/poorly at the start, and another of it playing the game well after training.

Iterations

Once I have this working, I have two disparate goals I can pursue. The first is to utilize the same tactics to create an agent in a different environment/gamespace. This would be a challenge because the convolutional network would have to be adjusted for the new screen, and further because the fully connected layers will have to be adjusted for the new action space. I think this is a reasonable/good stretch goal as it will demonstrate comfortability/familiarity with the code without being prohibitively difficult.

The second goal would be to work on implementing a more complicated algorithm for the agent to determine which action to take. I wouldn't be creating it myself, other algorithms currently exist/are used, but it would be an effort of trying to understand how they are actually implemented. I think this goal is harder, and should only be attempted if I have significant time left after getting the MVP finished.

Known Unknowns

The main difficulty for my project is that I don't have an authority with domain knowledge for this subject; I don't have someone who knows how to get this up and running themselves. This means my project is more exploratory/off the beaten path/independent.

Aside from this there are many known unknowns for me. I'm new to working in OpenAI's gym, I'm new to Keras/building neural networks, and new to using DQN/reinforcement learning algorithms.